

Görev-15

Amaç:

Bu görevde; mevcuttaki projelere iki tane daha WebAPI projesi ekleyerek e-Ticaret ve Admin MVC projelerinin bu API'leri kullanılması sağlanacaktır. Bu API projelerinden birisi DB CRUD işlemlerini gerçekleştirmek ve JWT oluşturmaktan, diğeri ise sadece dosya upload ve download işlemlerini yapmaktan sorumlu olacaktır. Böylelikle amacımız; Web API, JWT Authentication ve dosya upload/download işlemlerini gerçekleştirebilmek olacaktır.

Yapılacaklar:

Data API Projesi için yapılacak işlemler:

- Solution altına "App.Api.Data" isimli bir ASP.NET Core Web API projesi ekleyin. Görevi tamamladığınızda bu proje hem e-Ticaret hem de Admin MVC projelerindeki tüm veritabanı işlemlerini, hem de JWT oluşturma işini üstlenecek olup MVC projeleri doğrudan veritabanına erişemeyecek, veritabanı ve JWT işlemleri için bu API projesine istekte bulunacaktır.
- Data katmanını oluşturan class library projesini bu API projesine proje referansı olarak ekleyin.
- Hem e-Ticaret hem de Admin projelerindeki controller action'larını gözden geçirerek daha önceden oluşturmuş olduğumuz DataRepository üzerinden veritabanı işlemi yapan tüm action'lardaki tespit edin.
- Tespit ettiğiniz veritabanı işlemi yapan tüm kodlar için yeni eklediğiniz Data API projesine uygun API Controller'ları oluşturun (ör: OrderController, ProfileController gibi).
- Data API içerisinde oluşturduğunuz bu controller'lar içerisine uygun action'lar ekleyerek MVC projelerindeki tüm veritabanı işlemlerini tek tek ilgili API action'ına taşıyın.
- Her bir taşıma işlemi sonrasında MVC projelerinde ilgili veritabanı işlemi yapan yerde DataRepository kodlarını kaldırarak yerine .NET Framework'de dahili olarak bulunan HttpClient class'ını kullanın ve yeni oluşturduğunuz API action'ına uygun HTTP metodu ve URL ile (gerekliyse response body'yi de ekleyerek) HTTP isteğinde bulunun. Cevap olaran dönen response'dan status kod ve response body ile MVC action'larının öncekine benzer şekilde sonuçlanmasını sağlayın.
- Bu işlemler tamamlandığında tüm MVC action'larını tek tek deneyerek önceki durumuyla aynı şekilde davrandığını test edin.

JWT için yapılacak işlemler:

- Data API (*App.Api.Data*) projesi içerisindeki Program.cs dosyasında Bearer JWT Authentication altyapısı için gerekli değişiklikleri yapın.
- Data API içerisine “AuthController” isimli bir controller oluşturun.
- Bu controller içerisine e-mail ve password alıp kullanıcı varsa ilgili kullanıcının kişisel bilgilerini ve rollerini claim olarak içerek bir token oluşturarak response olarak döndüren bir action ekleyin. e-mail ve password ikilisinin uyduğu bir kayıt yoksa 404-NotFound sonucu döndürün.
- Bu action’ı “AllowAnonymous” attribute’u ile niteleyin.
- Token oluşturma işlemini Postman ile test edip gelen response içerisindeki token’ı jwt.io sitesinde deneyerek ayarladığınız claim’lere sahip olduğundan emin olun.
- Data API controller ve/veya action’larına aşağıda verilen **Yetkiler Tablosuna** uygun şekilde “Authorize” attribute’u ile rol tabanlı yetkilendirmeleri ekleyin.
- Postman aracılığı ile API’den önce token alıp ardından token sahibinin yetkili olduğu ve olmadığı endpoint’lere isteklerde bulunup beklenildiği gibi response döndürüldüğünden emin olun (Ör: 200-OK, 201-Created veya 401-Unauthorized)
- e-Ticaret ve Admin MVC projelerinden Data API projesine HttpClient ile yapılan isteklerde login olunduysa token’ın header içerisinde gönderilmesini sağlayın. Kılavuz olması için örnek bir senaryo aşağıdaki gibi olabilir:
 - a. e-Ticaret MVC’den login olduğunda Data API’nin login endpoint’ine e-mail ve şifre ile POST isteği gönderilmesi,
 - b. Gelen cevap 404-NotFound ise View’da “e-mail veya şifre yanlış” mesajının gösterilmesi,
 - c. Gelen cevap 200-OK ise response body’den JWT okunarak MVC response’unda Cookie olarak set edilmesi,
 - d. Bu noktada Data API’den token alınmış ve MVC aracılığı ile tarayıcıda cookie olarak tutuluyor durumda olacaktır. Bu cookie, tarayıcı tarafından her istekte MVC uygulamasına otomatik olarak gönderileceğinden her istek öncesi araya girilip, bu cookie okunup token olarak ayarlanması gerekmektedir. Böylelikle MVC projelerindeki Authentication altyapısı, fazla bir değişiklik yapmadan eskisi gibi çalışmaya devam edecektir. Sadece Cookie authentication’ının JWT authentication’a çevrilmesi ve JWT authentication konfigürasyonundaki *OnMessageReceived* event’i ile her istekte JWT içeren cookie’nin varsa okunup Token olarak ayarlanması gerekecektir. (Bkz: <https://spin.atomicobject.com/net-core-jwt-cookie-authentication/>)
 - e. MVC projesinden logout yapıldığında sadece ilgili cookie’nin (JWT içeren) silinmesi yeterli olacaktır.
 - f. Bu adımların ardından Data API’ye HttpClient ile istek gönderen her yerde cookie’deki bu token, request oluşturulurken “Authorization: Bearer {token}”

formatında request header'ına eklenecek şekilde deęişiklikler yapılmalıdır. Böylelikle Data API de aynı token ile sadece yetkilendirilmiş isteklere cevap verebilecektir.

- Data API uygulaması hem e-Ticaret hem de Admin MVC projeleri için aynı action ile token üreteceğinden sizden:
 - a. Seller ve Buyer rollerindeki kullanıcıların Admin projesine login olamaması,
 - b. Admin rolündeki kullanıcıların da e-Ticaret projesine login olamaması

için gerekli bir altyapı sağlamanız beklenmektedir.

File API Projesi için yapılacak işlemler:

- Solution altına “App.Api.File” isimli bir ASP.NET Core Web API projesi ekleyin. Görevi tamamladığınızda bu proje, dosyaları kaydetmekle sorumlu olacak, istenildiğinde dosyaları response olarak döndürecektir.
- Bu projeye “FileController” isminde bir controller ekleyin.
- FileController içerisine “Upload” isminde bir action oluşturun. Bu action HTTP POST metodu ile çalışmalıdır.
- Upload action'ında dosyaların “IFormFile” arayüzü ile alınıp daha önceden belirlediğiniz bir klasöre kaydedilmesini sağlayın. Kaydedilme başarılıysa 201-Created response kodu ile cevap döndürün.
- FileController içerisine “Download” isminde bir action oluşturun. Bu action HTTP GET metodu ile çalışmalıdır.
- Download action'ı Query String olarak dosya adı almalıdır. Buna göre upload action'ında kayıt yapılan klasörde bu isimde bir dosya varsa “return File(...)” ile dosyayı response ile göndermelidir. Bu isimde bir dosya yoksa 404-NotFound cevabı verilmelidir.
- e-Ticaret ve Admin projelerinde ürün görselleri ve (varsa) profil resimlerinin kaydedilmesi ve View'larda gösterilmesi için bu endpoint'ler kullanılmalıdır. Bunun için MVC projesinde gerekli deęişikleri yapın.
- Bu işlemlerin ardından satıcı (Seller) rolündeki bir kullanıcı ile bir ürüne görseller ekleyin. Ardından ürün sayfasında bu görsellerin File API'den alınarak görüntülenebildiğini test edin.

Yetkiler Tablosu (Görev-14'deki ile aynı):

No	Proje	Modül	User Story	Role: Buyer	Role: Seller	Role: Admin
1	e-ticaret	Auth	Kullanıcı sisteme kayıt olabilecektir	✓	✗	✗
2	e-ticaret, admin	Auth	Kullanıcı sisteme email ve şifre ile giriş yapabilecektir	✓	✓	✓
3	e-ticaret, admin	Auth	Kullanıcı sistemden çıkış yapabilecektir	✓	✓	✓
4	e-ticaret, admin	Profile	Kullanıcı kendi profil sayfasını görüntüleyebilecektir	✓	✓	✓
5	e-ticaret	Profile	Kullanıcı, satıcı olmak isteğini talep edebilecektir	✓	✗	✗
6	e-ticaret	Cart	Kullanıcı stokta varsa ürünü sepetine ekleyebilecektir	✓	✓	✗
7	e-ticaret	Cart	Kullanıcı ürünü sepettinden çıkarabilecektir	✓	✓	✗
8	e-ticaret	Cart	Kullanıcı sepetteki ürününün adedini güncelleyebilecektir	✓	✓	✗
9	e-ticaret	Cart	Kullanıcı sepetinin detaylarını görüntüleyebilecektir	✓	✓	✗
10	e-ticaret	Order	Kullanıcı ödeme yaptıktan sonra adres bilgisi girecek, ardından siparişi otomatik oluşacaktır	✓	✓	✗
11	e-ticaret	Order	Kullanıcı siparişlerinin listesini görebilecektir	✓	✓	✗
12	e-ticaret	Order	Kullanıcı tek bir siparişinin detaylarını görebilecektir	✓	✓	✗
13	admin	Category	Kullanıcı sisteme yeni bir kategori ekleyebilecektir	✗	✗	✓
14	e-ticaret	Category	Kullanıcı bir kategori altındaki ürünleri listeyebilecektir	✓	✓	✗
15	e-ticaret	Product	Kullanıcı sisteme yeni ürün ekleyebilecektir	✗	✓	✗
16	e-ticaret	Product	Kullanıcı satıştaki kendi ürünlerini listeleyebilecektir	✗	✓	✗
17	e-ticaret	Product	Kullanıcı sistemdeki mevcut kendi ürününün fiyatını güncelleyebilecektir	✗	✓	✗
18	e-ticaret	Product	Kullanıcı sistemdeki mevcut kendi ürününün stoğunu güncelleyebilecektir	✗	✓	✗
19	e-ticaret	Product	Kullanıcı sipariş oluşturduktan sonra ürüne yorum ve yıldız verebilecektir	✓	✓	✗
20	e-ticaret	Product	Kullanıcı tek bir ürünün detaylarını görebilecektir	✓	✓	✗
21	e-ticaret	Product	Kullanıcı genel ürün araması yapabilecektir	✓	✓	✗
22	e-ticaret, admin	Product	Kullanıcı, ürünü pasif hale getirebilecektir	✗	✓	✓
23	e-ticaret	ProductComment	Kullanıcı satın aldığı ürünlere yorum ve yıldız verebilecektir	✓	✓	✗
24	admin	ProductComment	Kullanıcı, ürün yorumlarını herkesin görebilmesi için onaylayabilecektir	✗	✗	✓
25	admin	User	Kullanıcı, sistemdeki tüm kullanıcıları listeleyebilecektir	✗	✗	✓
26	admin	User	Kullanıcı, diğer kullanıcıları aktif/pasif yapabilecektir	✗	✗	✓
27	admin	User	Kullanıcı, satıcı olma taleplerini onaylayabilecektir	✗	✗	✓

Teslim: Proje klasörünü zip'leyip classroom'daki Görev-15 paylaşımına ek olarak ekleyin.